# Umgang mit Wrapperklassen (Parsing, Boxing, Strings)

\sic-juc-3a\Unterricht\src\\_20241129WrapperCasten\DungeonCrawl Tag 5 3a JuC.pdf

<https://freiheit.f4.htw-berlin.de/prog2/wrapper/>

Auszüge und Ableitungen aus dieser Quelle:

**Erzeugung von Wrapper-Objekten (aus primitiven Werten oder Strings)**

Bei Handhabung der Wrapperklasse Character (char) kann kein String übergeben werden.

Klassischer Konstruktor ist veraltet (depricated), man nehme nun die jeweils statische Methode der Wrapperklasse valueOf(). Diese kann per String oder Literal bedient werden.  
Da die valueOf() Methode ein neues Objekt liefert, nennt man sie auch Erzeugermethode.

Integer iObject2 = new Integer(100); oder Integer iObject4 = new Integer("100");

Integer iObject1 = Integer.valueOf(100); oder Integer iObject2 = Integer.valueOf("100");

Alternativ kann man mit Autoboxing primitive Werte implizit in Wrapperobjekte wandeln.

Integer io1 = 100; Boolean bo1 = true; Double do1 = 5.5;

**Extrahieren von primitiven Werten (aus Wrapperobjekten oder Strings)**

Um den Wert aus der Wrapperklasse zurückzuerhalten kann man entweder das Autounboxing verwenden oder die xxxValue()-Methode zur Zuweisung eines primitiven Wertes verwenden; xxx steht hierbei für den primitiven Datentyp wie zum Beispiel int.

XXX (Wrapper) wO = val; xxx var = wO.xxxValue(); // value

Integer io1 = 100; int i1 = io1.intValue(); // 100

Boolean bo1 = true; boolean b1 = bo1.booleanValue(); // true

Character co1 = 'a'; 'a'; char c1 = co1.charValue(); // c

Um aus einem String die Werte zurückzuerhalten nutzt man hingegen die String.parseXxx()-Methode. Also zum Beispiel:

String s = “10”; 🡪 **int i** = String.parseInt(s); //(i = 10), nicht Integer i !

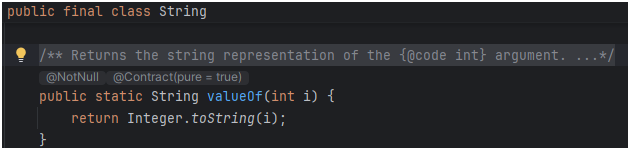
Übrigens erben die numerischen Datentypen alle von der Klasse Number. Diese enthält die Objektmethoden **byteValue(), doubleValue(), floatValue(), intValue(), longValue() und shortValue().** Deswegen werden in allen numerischen Wrapper-Klassen auch alle diese Objektmethoden zur Verfügung gestellt. **[Diese extrahieren den primitiven Wert.]**

**Umwandlung in einen String (aus Wrappern oder primitiven Werten)**

Hier haben wir 4 verschiedene Möglichkeiten.

1. Wir nutzen die Wrapperklasse statisch mit Literal: String s = XXX.toString(xxx)
2. Wir nutzen die Stringklasse statisch mit Literal: String s = String.valueOf(xxx)
3. Die toString()-Methode der Wrapperinstanz. String s = objectXXX.toString()
4. Man nutze die Stringklasse statisch mit Wrapper: String s = String.valueOf(XXX)

Übrigens nutzt String (durchreichend) die valueOf-Methoden der Wrapper je Parameter.



**Weiteres und Anmerkungen**

Zumeist wird ohnehin implizit in String konvertiert und auch das Auto(-un)-Boxing ist komfortabler. Für die Praxis ist das also alles halb so wild, aber trotzdem gut zu wissen.

Übrigens prüft die Wrapperklasse die Gleichheit (equals()) durch Downcasting des Übergebenen Objektes und wendet die xxxValue-Methode darauf an um direkt die Literale mit == zu vergleichen. Natürlich nur, wenn das Objekt eine Instanz davon ist.   
Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

**Zusammenfassung**

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

**Versuch einer Lernzielkontrolle (dieser Versuch: 100% (aber war ne schwere Geburt))**

|  |  |  |  |
| --- | --- | --- | --- |
| **von (↓) / zu (→)** | **zu primitiv (xxx)** | **zu Wrapper (XXX)** | **zu String** |
| **von primitiv (xxx)** |  | Autoboxing:  XXX wVar = primVar;  statische valueOf()-Methode: XXX wVar = XXX.valueOf(xxx)  (veraltet): XXX wVar = new XXX(xxx) | Statisch über Wrapperklasse mit toString: String s = XXX.toString(xxx)  Statisch mit der Stringklasse über valueOf():  String s = String.valueOf(xxx) |
| **von Wrapper (XXX)** | Autounboxing: xxx var = objectXXX;  xxxValue()-Methode:  xxx var = objectXXX.xxxValue(); |  | Über das Wrapperobjekt mit toString()  String s = varXXX.toString()  Statisch mit der Stringklasse über valueOf():  String s = String.valueOf(XXX) |
| **von String** | String.parseXxx()-Methode: xxx var = String.parseXxx(s) | String.valueOf()-Methode (benutzt die XXX.valueOf-Methode)  XXX wVar = String.valueOf(“xxx”)  (veraltet): XXX wVar = new XXX(“xxx”) |  |

**Vorlagentabelle für Lernzielkontrolle**

|  |  |  |  |
| --- | --- | --- | --- |
| **von (↓) / zu (→)** | **primitiv (xxx)** | **Wrapper (XXX)** | **String** |
| **primitiv (xxx)** |  |  |  |
| **Wrapper (XXX)** |  |  |  |
| **String** |  |  |  |